ESSVCS: An Enriched Secre Sharing Visual Cryptography

Feng Liu*, Wei Q. Yan**, Peng Li***, Chuankun Wu*
*State Key Laboratory of Information Security, Institute of Information Engineering
Chinese Academy of Sciences, Beijing 100093, China
**School of Computing and Mathematical Sciences,
Auckland University of Technology, 1142, New Zealand
***Department of Mathematics and Physics,
North China Electric Power University, Baoding, Hebei, 071003 China
e-mail: fengliu.cas@gmail.com

Abstract. Visual Cryptography (VC) is a powerful technique that combines the notions of perfect ciphers and secret sharing in cryptography with that of raster graphics. A binary image can be divided into shares that are able to be stacked together so as to approximately recover the original image. VC is a unique technique in the sense that the encrypted message can be decrypted directly by the Human Visual System (HVS). The distinguishing characteristic of VC is the ability of secret restoration without the use of computation. However because of restrictions of the HVS, pixel expansion and alignment problems, a VC scheme perhaps can only be applied to share a small size of secret image. In this paper, we propose a general method to let the VC shares carry more secrets, the technique is to use cypher output of private-key systems as the input random numbers of VC scheme, meanwhile the encryption key could be shared, the shared keys could be associated with the VC shares. After this operation, VC scheme and secret sharing scheme are merged with the private-key system. Under this design, we implement a (k, t, n)-VC scheme. Compared to those existing schemes, our approach could greatly enhance the ability of current VC schemes and could cope with pretty rich secrets.
Keywords: Secret Sharing, Visual Cryptography, Covert Data.

1 Introduction

Visual Cryptography Scheme (VCS) was firstly introduced by Naor and Shamir [1], which shares a secret image into $n \in Z$ pieces (printed on transparencies). The merit of VCS is that the decoding process is computation-free. The original image is able to be recovered by stacking any $k \leq n$ shares transparently. The underlying operation of the stacking is the logic *OR*. Lots of research focused on the novel applications of VCS [2–5]. Recently, some books covered an extensive range of topics related to VCS [6–8].

In traditional VCS, the amount of secret is severely constrained: pixel expansion of the shares implies that the size of secret image cannot be too big, because a big transparency is inconvenient for the shares alignment; human eyes can only identify patterns of secret image when the contrast is good enough, i.e. the lines and dots in the patterns should be a block of pixels rather than a single pixel; Because of the alignment problem [9, 10], pixels within the shares cannot be too small. Many studies tried to increase the secret volume of VC shares, such as sharing a plural number of secret images in one VCS [11, 12], using rotated shares [13] or using color VC scheme [14–16]. However, these methods could not increase the capability too much if the ratio $R = \frac{t}{m}$ is taken into consideration, where t is the number of secret bits that are shared by every m sub-pixels. For the color VC scheme, it usually degrades quality of the revealed secret image severely. In this paper, we measure the capability of VC scheme by using the secret bits that will be shared.

The main contribution of this paper is that, the random inputs of VC scheme could be applied to carry covert data, the ciphertext of those private-key system based encryption algorithms could be considered as random inputs of a VC scheme, hence it increases the amount of secret shared by VCS. By using Shamir's secret sharing scheme [17], the encryption key is able to be shared into n sub-keys that could be associated with the corresponding shares. We call this scheme as the Enriched Secret Sharing VC Scheme (ESSVCS), or 3-in-1 VCS. The scheme articulately combined

the two secret sharing schemes and private-key encryption scheme together. The secret shared by the ESSVCS includes two parts: secret and covert data. Figure 1 and Figure 2 illustrate the encryption and decryption procedures. The reasonabilities, possiblities and potential problems will be discussed in the following sections of this paper.

In Figure 1, we encrypt a plaintext $S_{plaintext}$ by using the key S_{Key} so as to generate the ciphertext $S_{ciphertext}$ via the function $En(S_{Key}, S_{plaintext})$, the ciphertext $S_{ciphertext}$ could be used as a VCS share to split a visual secret S_1 by using (k, n)-VCS scheme, so as to get the visual shares $V_1, V_2, \cdots, V_n$; on the other hand, the key S_{Key} could be shared using the Polynomia-based Secret Sharing Scheme (PSSS) namely (t, n)-PSSS to get the sub-keys $SK_1, SK_2, \cdots, SK_n$, we could convert the sub-keys to binary images, so we could get the imagelets $I_1, I_2, \cdots, I_n$; we now concatenate the imagelets $I_1, I_2, \cdots, I_n$ and the visual shares $V_1, V_2, \cdots, V_n$ together to get the visual shares $S_1, S_2, \cdots, S_n$.

In Figure 2, we decrypt the corresponding secrets in Figure 1 by using the n shares $S_1, S_2, \cdots, S_n$, we superimpose the n shares and get the secret S_1, we extract data from any t out of n shares so as to get the secret $S_{cipertext}$ and t sub-keys $SK_1, SK_2, \cdots, SK_n$, we use the Lagrange's algorithm to interpolate the key S_{Key}, the key S_{Key} and the cipertext $S_{cipertext}$ work together to decrypt the plaintext $S_{plaintext}$ by using the function $De(S_{Key}, S_{cipertext})$. The reasons to guarantee this step will be explained in the following sections of this paper.

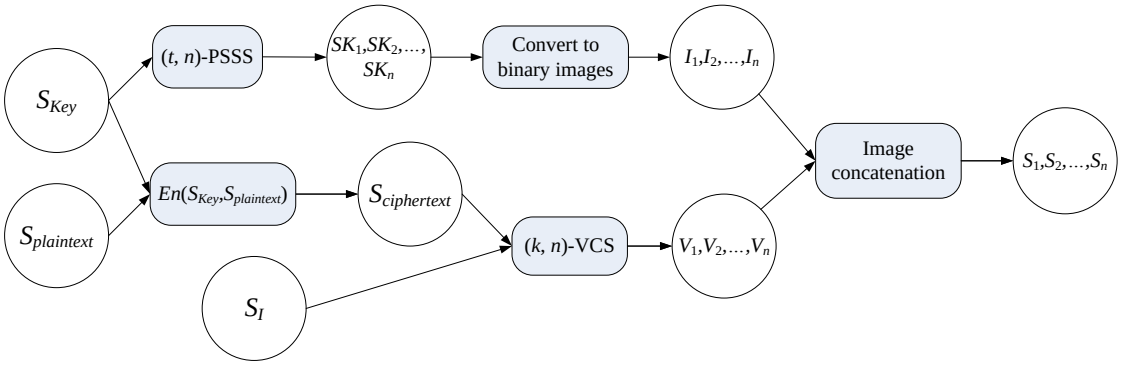

Fig. 1: The encryption process of the (k, t, n)-ESSVCS

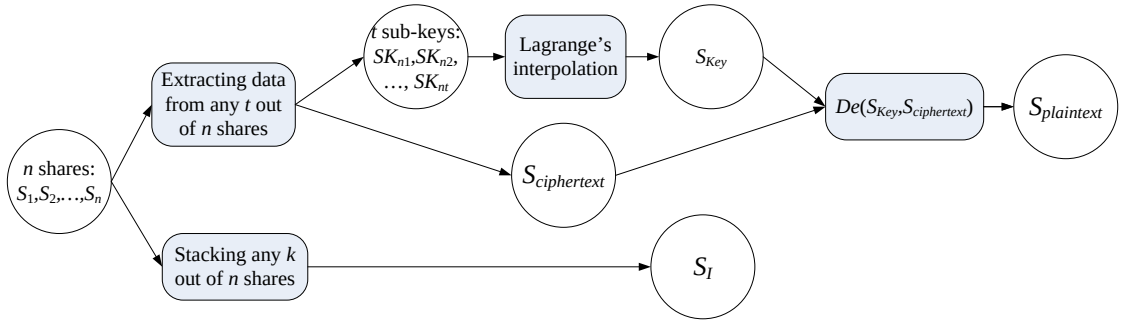

Fig. 2: The decryption process of the (k, t, n)-ESSVCS

In order to share the covert data, we need employ computational devices. By utilizing a (k, t, n)-ESSVCS ($k \leq t$), the VC scheme that carries additional covert data where any k out of n participants can visually recover the secret by stacking the shares, any t out of n participants can restore the additional covert data by computation. There are two computer aided VCS's schemes [18, 19], one is called 2-in-1 Image Secret Sharing Scheme (TiOISSS) [18, 19], which is able to reveal a secret image by stacking the shares and restore a much finer gray image by computation. Li et al. [20] improved

Yang et al.'s TiOISSS [19] by gray mixing model. The comparisons between our ESSVCS and the TiOISSS will be given in Section 4. Fang et al. [5, 21] also tried to make use of the pseudo-random inputs to carry confidential data. Unfortunately, the scheme is only for (2,2) access structure.

The proposed (k, t, n)-ESSVCS in this paper is a multi-threshold secret sharing scheme. k out of n members can share one secret, whereas a majority of participants $t \leqslant n$ can access the additional secret. By comparing our ESSVCS and any 2D encoding methods, we find that decoding a secret totally relies on a computing device by using any 2D encoding methods. If participants are in the scenario where there is no such computing devices, they cannot extract any information. But with our ESSVCS scheme, the participants could stack the shares and get part of the secret. Hence, our proposed ESSVCS scheme will have much wider application.

In this paper, we propose a specific construction of general (k, t, n)-ESSVCS by taking the VCS proposed in [12] and secret sharing scheme [17] into consideration. We investigate some relevant issues of ESSVCS scheme such as pseudo-random numbers as the input of VCS scheme, sufficient conditions to uniquely determine a share, and secret capacity of the proposed ESSVCS scheme; we also proposed an efficient decoding algorithm, comparisons to other schemes will be presented at last.

This paper is organized as follows. In section 2, we will give some preliminary results. In section 3, we will propose a general construction of the ESSVCS based on the construction of VC scheme in [12] and point out some relevant issues of ESSVCS. In section 4, we will compare the proposed scheme with the TiOISSS scheme. Finally, we will draw our conclusion in section 5.

2 Preliminaries

In this section, we will present some definitions about VC scheme, introduce the Droste's construction of (k, n)-VC Scheme [12] and Shamir's secret sharing scheme, namely Polynomial-based Secret Sharing Scheme (PSSS) [17], they are the start point of our proposed scheme.

2.1 VCS

We restrict ourselves to the images only consisting of black and white pixels, where we denote by '1' for a black pixel and '0' for a white pixel. In this paper, we only take the threshold (k, n)-VCS into consideration. For a vector $v \in GF^m(2)$, we denote by $w(v)$ as Hamming weight of the vector v. A (k, n)-VCS, denoted by (C_0, C_1), consists of two sets (pairwise different collection) of $n \times m$ Boolean matrices C_0 and C_1. To encrypt a white (*resp.* black) pixel, a dealer (the one who sets up the system) randomly chooses one of the share matrices (in the practical sense, the dealer can only choose the share matrices pseudo-randomly) from C_0 (*resp.* C_1) and distributes its rows (shares) to the n participants. More precisely, we present a formal definition of the (k, n) - VCS as follows.

Definition 1. *Let k, n, m, l and h be non-negative integers satisfying $2 \leq k \leq n$ and $0 \leq l < h \leq m$. The two sets of $n \times m$ Boolean matrices (C_0, C_1) constitute a (k, n)-VCS if the following properties are satisfied:*

1. (**Contrast**) *For any $s \in C_0$, the OR of any k out of the n rows of s, is a vector v that, satisfies $w(v) \leq l$.*
2. (**Contrast**) *For any $s \in C_1$, the OR of any k out of the n rows of s, is a vector v that, satisfies $w(v) \geq h$.*
3. (**Security**) *For any $i_1 < i_2 < \cdots < i_t$ in $\{1, 2, \cdots, n\}$ with $t < k$, the two collections of $t \times m$ matrices F_j for $j \in \{0, 1\}$, obtained by restricting each $n \times m$ matrix in C_j to rows $i_1, i_2, \cdots i_t$, are indistinguishable in the sense that they contain the same matrices with the same frequencies.*

In the above definition, m is called pixel expansion of the shares. A pixel of the original secret image is represented by m sub-pixels in the recovered secret image. In general, we are interested in schemes with m being as small as possible.

In Definition 1, the first two properties ensure that any k participants will be able to distinguish the black and white pixels, and the third property ensures security of the scheme that any $k-1$ or fewer participants can gain no information about content of the secret.

In order to share a complete image, the scheme has to be applied to all the pixels in the image. In the traditional VCS, the secret sharing method is applied to the secret pixels one at each time. However, we extend this method to share q secret pixels at each time, and call this scheme as the *q-pixel encryption model*. The traditional model is the *1-pixel encryption model*. The difference between the *1-pixel encryption model* and the *q-pixel encryption model* is that: in the *1-pixel encryption model*, the dealer generates one pseudo-random number which guides the choice of a share matrix at each time. However, in the *q-pixel encryption model*, the dealer generates one pseudo-random number which guides the choice of q share matrices at each time.

Now let's take VCS into consideration, the C_0 and C_1 are constructed from a pair of $n \times m$ matrices M_0 and M_1, which are called *basis matrices*. The set C_i $(i = 0, 1)$ consists of the matrices obtained by permuting all the columns of M_i. This approach of VCS construction will have small memory requirements (it only keeps the basis matrices) and high efficiency (to choose a matrix in C_0 (*resp.* C_1) as it only needs to generate a permutation of the basis matrix). When the set of a VCS C_0 (*resp.* C_1) can be generated by the basis matrix, we call such VCS as the *basis matrix VCS*. Many studies in the literatures proposed to construct the *basis matrix VCS*, such as [12, 22, 23].

Recall that, by definition, the share matrices in C_0 (*resp.* C_1) are pairwise differently. Denote the different columns in the basis matrix M_i as $c_1, c_2, \cdots, c_e$ and the multiplicities of these columns are $a_1, a_2, \cdots, a_e$, we have that the number of share matrices in C_i is $|C_i| = \frac{(\sum_{i=1}^{e} a_i)!}{\prod_{i=1}^{e} a_i!}$, for $i \in \{0, 1\}$ (these share matrices are pairwise different). In order to choose a share matrix in C_i pseudo-randomly, length of the pseudo-random input for one secret pixel should be at least $\log_2 |C_i|$ bits.

2.2 Droste's construction of (k, n)-VCS

In this paper, we take the construction of Droste [12] as our building block, and we recall his construction as follows:

Droste's Construction of (k, n)-VCS proposed in [12]:

Setup Let M_0 and M_1 be two empty matrices, where the basis matrices M_0 and M_1 are considered as the collections of their columns;

step 1 For all even $p \in \{0, 1, \ldots, k\}$, call **ADD($p$,$M_0$)**;

step 2 For all odd $p \in \{0, 1, \ldots, k\}$, call **ADD($p$,$M_1$)**;

step 3 Define P_0 (resp. P_1) be the collection consisting of all columns of every restriction of k rows of M_0 (resp. M_1), and define S_0 (resp. S_1) be the set consisting of all k-length boolean columns with an even (resp. odd) number of 1's. Define the remaining of M_0 (resp. M_1) be $P_0 \backslash S_0$ (resp. $P_1 \backslash S_1$), and define the *rest* of M_0 (resp. M_1) be the columns in the remaining of M_0 (resp. M_1), but not in the remaining of M_1 (resp. M_0), i.e. the *rest* of M_0 is $\{P_0 \backslash S_0\} \backslash \{P_1 \backslash S_1\}$ and the *rest* of M_1 is $\{P_1 \backslash S_1\} \backslash \{P_0 \backslash S_0\}$. If the *rests* are not empty:
(a) If p is an even number, add to M_0 all columns adjusting the *rest* of M_1 by calling **ADD(p,M_0)**, where p is the number of 1's in column $l \in \{P_1 \backslash S_1\} \backslash \{P_0 \backslash S_0\}$.
(b) If p is an odd number, add to M_1 all columns adjusting the *rest* of M_0 by calling **ADD(p,M_1)**, where p is the number of 1's in column $l \in \{P_0 \backslash S_0\} \backslash \{P_1 \backslash S_1\}$.

where the subroutine ADD is: **ADD(p,M)**

1 If $p \leq k - p$, add every column with $q = p$ (1's to M).
2 If $p > k - p$, add every column with $q = p + n - k$ (1's to M).

Example 1. (The Droste's construction of (3,4)-VCS) In the first step, every column with zero and three 1's is added to M_0. In the second step, every column with one and four 1's is added to M_1. The generated M_0 and M_1 are shown as follows,

$$M_0 = \begin{bmatrix} 01110 \\ 01101 \\ 01011 \\ 00111 \end{bmatrix}, M_1 = \begin{bmatrix} 10001 \\ 01001 \\ 00101 \\ 00011 \end{bmatrix}$$

Now every restriction of M_0 (resp. M_1) contains every even (resp. odd) column and besides that every column with three (resp. zero) 1's. So in the first run of step 3, every column with zero (resp. 4) 1's is added to M_0 (resp. M_1). After that, we have new M_0 and M_1, and the rests of M_0 and M_1 are empty. The final basis matrices M_0 and M_1 are shown as follows,

$$M_0 = \begin{bmatrix} 011100 \\ 011010 \\ 010110 \\ 001110 \end{bmatrix}, M_1 = \begin{bmatrix} 100011 \\ 010011 \\ 001011 \\ 000111 \end{bmatrix}$$

□

2.3 PSSS

Shamir[17] introduced the (t, n)-PSSS ($t \leq n$) to share the secret data into n shares. Any t shares can be used to reconstruct the secret, but any $t-1$ or less shares get no information about the secret. To share the secret, it randomly generates a $(t-1)$-degree polynomial using modular arithmetic:

$$f(x) = (a_0 + a_1 x + \ldots + a_{t-1} x^{t-1}) \mod p \tag{1}$$

where a_0 is replaced by the secret data, p is a prime number greater than a_0 and n. The coefficients $a_1, a_2, \ldots, a_{t-1}$ are randomly chosen from a uniform distribution over the integers in $[1, p)$. Then we could generate n shares $(x_i, f(x_i)), i = 1, 2, \ldots, n$. Later, with any t out of the n shares, we can uniquely determine a $(t-1)$-degree polynomial as follows,

$$f(x) = \sum_{j=1}^{t} f(j) \prod_{i=1, i \neq j}^{t} \frac{x-j}{j-i} \tag{2}$$

Particularly, the coefficient a_0 of the polynomial $f(x)$ is decrypted (Lagrange's interpolation). However, any $t-1$ or fewer shares cannot uniquely determine a $(t-1)$-degree polynomial. Hence no information about the secret is revealed.

Example 2. (The Shamir's $(2, 3)$-PSSS) In a $(2, 3)$-PSSS, the prime number p is chosen as 251. Let the secret number be 45, which is in the range of $(0, p-1)$. In the sharing process, the secret number 45 replaces the constant coefficient of a 1-degree polynomial, and another coefficient, for example 145, is randomly chosen in $(1, p-1)$. Therefore, we can generate a 1-degree polynomial as follows,

$$f(x) = (45 + 145x) \mod 251$$

Then we can generate 3 shares $(x_i, f(x_i))$, where x_i is the ID of the i-th participant. Without loss of generality, let i be the ID of the i-th participant, we have three shares (1, 190),(2, 84) and (3, 229).

In the revealing process, any two out of three shares can uniquely determine a 1-degree polynomial by Equation (2). Finally, the secret number 45 can be decrypted. □

3 ESSVCS

In this section, we first propose a construction of the ESSVCS scheme by taking the pseudo-random inputs as a sub-channel, and then study some relevant issues of the ESSVCS: 1) The pseudo-randomness that the input of VCS requires; 2) The sufficient conditions to uniquely determine a share matrix in the set C_i for $i = 0, 1$; 3) The bandwidth of the sub-channel; 4) The method to decode the ciphertext of ESSVCS scheme.

3.1 Construction of general (k, t, n)-ESSVCS

The main idea of this proposed scheme is to treat the private-key encryption algorithm as the pseudo-random generator of VCS. Thus the VCS can naturally carry the additional covert data encrypted by the private-key algorithm. In this paper, we take the VCS proposed in [12] as the building block. In practical, the encryption algorithm can be the AES or Twofish, etc. The cipher block chaining (CBC) [24] encryption mode is employed. The encryption key S_{Key} in ESSVCS is shared by (t, n)-PSSS into n sub-keys $SK_1, SK_2, \ldots, SK_n$. Therefore, any t or more sub-keys could be used to reveal the secret key, while any $t-1$ or less sub-keys together could restore the secret key.

Before showing the construction, we need present the assumption that participants know the access structure they belong to, i.e. the i-th participant knows by himself/herself that (s)he is the i-th participant. Usually, the access structure of a VCS is not one part of secret, therefore this assumption is reasonable.

Construction 1
Encryption process of (k, t, n)-ESSVCS:
Input: *The secret image S_I, covert data $S_{Plaintext}$ and the secret key S_{Key}.*
Output: *n shares.*
Step 1: *Encrypt the covert data $S_{plaintext}$ by using the key S_{Key}, $S_{ciphertext} = En(S_{Key}, S_{plaintext})$;*
Step 2: *Share the secret image S_I into n shares $V_1, V_2, \ldots, V_n$ by using the (k, n)-VCS, where the encrypted data from the Step 1 is employed as the pseudo-random input of the (k, n)-VCS;*
Step 3: *Share S_{Key} into n sub-keys $SK_1, SK_2, \ldots, SK_n$ by using (t, n)-PSSS, then convert these sub-keys into binary images $I_1, I_2, \ldots, I_n$, and concatenate I_i $(i = 1, 2, \ldots, n)$ with share V_i to get the final share S_i.*

Decryption process of (k, t, n)-ESSVCS:
Input: *Any t shares where $k \leq t$.*
Output: *The secret image S_I and the covert data $S_{plaintext}$.*
Step 1: *Stack any k shares to get the recovered secret image S_I;*
Step 2: *Determine the share matrices which are used to encrypt the secret image for each pixel by t shares, and hence get the ciphertext $S_{ciphertext}$;*
Step 3: *Extract t sub-keys from t shares, then reconstruct the secret key S_{Key} by Lagrange's interpolation.*
Step 4: *Decrypt the ciphertext.*
$S_{ciphertext}$ by using the S_{Key}, $S_{plaintext} = De(S_{Key}, S_{ciphertext})$.

Remarks:

In practical, key length of the AES or Twofish scheme, usually, is 128 bits. Therefore, each sub-key is generated and converted into a 128 bits binary image which only takes a small area in the share.

For the (k, t, n)-ESSVCS, by stacking k shares we can reconstruct the secret image S_I. If one obtains t rows, (s)he can uniquely determine a share matrix and hence obtain the ciphertext, where "can uniquely determine a share matrix" means that there only exists one share matrix in C_i

$(i = 0, 1)$ that contains these t rows (and "cannot uniquely determine" means there exist more than one share matrices that contain these t rows, hence we cannot determine which one is chosen by the dealer when encrypting the secret pixel). In another word, in order to get the ciphertext one needs t shares.

Security of the (k, t, n)-ESSVCS is based on the security of the encryption algorithm and that of VCS and PSSS scheme. Particularly, if an hacker wants to know the secret image, (s)he needs at least k shares; if (s)he wants to know the covert data encrypted by the encryption algorithm, (s)he needs at least t shares to extract the ciphertext and the secret key.

The VCS requires pseudo-random number inputs to guide the choice of VC share matrices. Denote the share matrices in C_i as $S_0^i, \cdots, S_{|C_i|-1}^i$ and $P(S_j^i)$ for $i = 0, 1$ and $j = 0, 1, \cdots, |C_i| - 1$ as the probability choosing the share matrix S_j^i. Hence inputs of the pseudo-random numbers should guarantee that

$$P(S_0^i) = P(S_1^i) = \cdots = P(S_{|C_i|-1}^i) \tag{3}$$

In order to choose a share matrix pseudo-randomly in C_i, the dealer needs at least $\log_2 |C_i|$ bits pseudo-random numbers (we will take the case that $\log_2 |C_i|$ is not an integer into consideration). Denote $B(j)$ as the binary representation of integer j with length $\log_2 |C_i|$, i.e. $B(j)$ is the binary string that represents j. Without loss of generality, we assume that when the pseudo-random number input is $B(j)$, the dealer chooses the share matrix S_j^i to encrypt the secret pixel i. Denote $P(B(j))$ as the probability of generating the binary string $B(j)$. According to the equation (3), we have:

$$P(B(0)) = P(B(1)) = \cdots = P(B(|C_i| - 1)) \tag{4}$$

In fact, ciphertext of AES or Twofish satisfies the equation (4), because they have passed the serial test [25]. Therefore, we can take AES or Twofish as the pseudo-random generator. This also is the ground truth why we do not use the covert data directly to guide the generation of shares.

To make things simple and clear, we give the following example for $(2, 2, 2)$-ESSVCS:

Example 3. The sets of share matrices of $(2, 2, 2)$-ESSVCS are as follows:

$$C_0 = \left\{ \begin{bmatrix} 10 \\ 10 \end{bmatrix}, \begin{bmatrix} 01 \\ 01 \end{bmatrix} \right\} \text{ and } C_1 = \left\{ \begin{bmatrix} 10 \\ 01 \end{bmatrix}, \begin{bmatrix} 01 \\ 10 \end{bmatrix} \right\}$$

The principle of choosing share matrix is that: if the pseudo-random input is 0, we choose the first share matrix in C_0 or C_1; if the pseudo-random input is 1, we choose the second option. Figure 3 presents an illustration for the procedure of the $(2, 2, 2)$-ESSVCS.

In Figure 3, a secret image having 64×128 pixels is split into Share 1 and Share 2. Size of the shares and the recovered secret image is 129×128. Since the length of each sub-key is 128 bits, it only takes one line at the bottom of each share to attach the sub-keys. Length of the ciphertext $S_{ciphertext}$ encrypted in the shares is 2^{13} bits, i.e. the sub-channel can be used to carry extra 2^{13} bits of covert data. In the first step of the reconstruction, the secret image can be visually revealed by stacking two shares. In the second step, two sub-keys SK_1 and SK_2 are extracted from the last row of two shares, and then we restore the secret key S_{Key} by Lagrange's interpolation. With further observation, the ciphertext can be obtained by the uniquely determined share matric by share blocks. For example, the first block of share 1 is constituted by two sub-pixels '0' and '1', and the first block of share 2 is also constituted by two sub-pixels '0' and '1'. Therefore, we can determine the share matrix, which is the second share matrix C_0 and the recovered ciphertext is '1'. Finally, we get the covert data $S_{plaintext}$ by decrypting the ciphertext $S_{ciphertext}$. □

3.2 Uniquely determine a share matrix

For the (n, n)-VCS, if one has all the n shares, (s)he can uniquely determine the share matrices used when sharing the secret image S_I and hence to know the ciphertext.

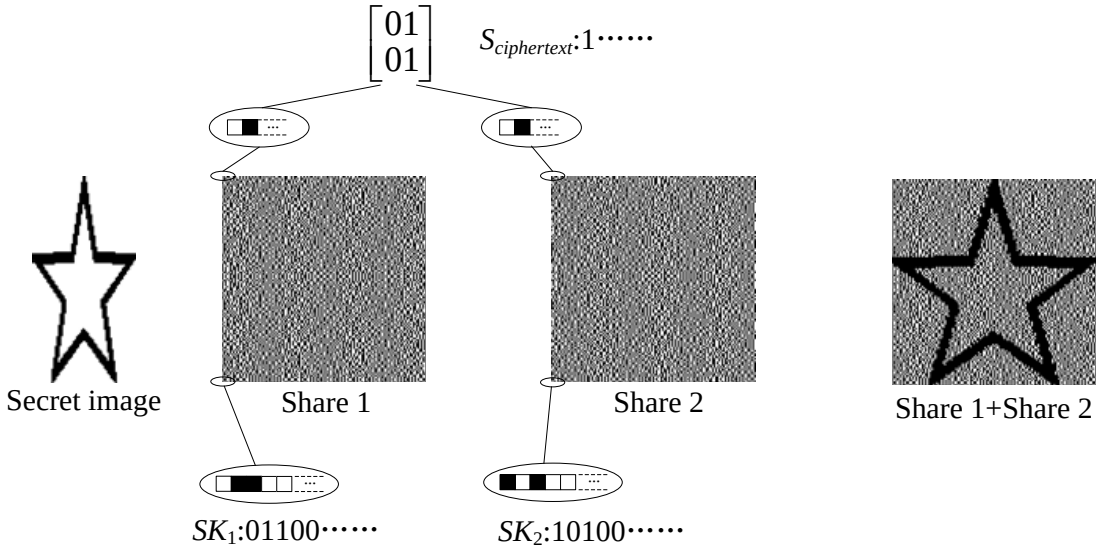

Fig. 3: The procedure of the $(2, 2, 2)$-ESSVCS

We then focus our discussion on the (k, n)-VCS with $k < n$: we find that, for the VCS in section 2, $n-1$ rows can uniquely determine a share matrix in the set C_0 (resp. C_1). The following theorem shows this result:

Theorem 1. *Denote M_0 and M_1 be the basis matrices constructed by (k, n)-VCS in [12], and denote C_0 and C_1 be the sets of share matrices generated from M_0 and M_1, respectively. If every t rows of a share matrix in C_i $(i = 0, 1)$ can uniquely determine a share matrix in C_i, then $t \geq n-1$.*

Proof: *First*, for the case of $t = n$, it obviously can uniquely determine an n-row matrix from its all n rows.

Second, we show any $n-1$ rows can uniquely determine a share matrix. According to the construction in [12], number of the 1's of each column in the basis matrix M_0 is from the set $T_0 = \{a|0 \leq a \leq \lfloor \frac{k}{2} \rfloor, a \mod 2 = 0\} \bigcup \{a + n - k | \lfloor \frac{k}{2} \rfloor < a \leq k, a \mod 2 = 0\}$, and number of 1's of each column in the basis matrix M_1 is from the set $T_1 = \{a|0 \leq a \leq \lfloor \frac{k}{2} \rfloor, a \mod 2 = 1\} \bigcup \{a + n - k | \lfloor \frac{k}{2} \rfloor < a \leq k, a \mod 2 = 1\}$. Hereafter, $\lfloor x \rfloor$ is the largest integer that is no greater than x and $\lceil x \rceil$ is the smallest integer no less than x.

Because $k < n$, when one has $n-1$ rows of a share matrix M, he can stack k shares and hence knows the secret pixel. Without loss of generality, suppose the secret pixel is black. We determine last row of the share matrix M as follows: for the column p_i of M, where $i \in \{1, \cdots m\}$, denote number of 1's of the $n-1$ rows in column p_i as h, then we have the entry of the last rows of column p_i be 0 if $h \in T_1$ and be 1 if $h+1 \in T_1$. Hence, the last row can be uniquely determined by the $n-1$ rows, because the participants know the access structure they belong to, the share matrix will be uniquely determined.

Third, we prove any $n-2$ rows cannot uniquely determine a share matrix. Consider the construction in [12], we have that the basis matrix M_1 contains all the columns with Hamming weight are equal to 1. Let A be a share matrix in C_1. Without loss of generality, there exist two different columns c_1 and c_2 in A, whose Hamming weights are equaled to 1. Denote the position of 1 in column c_1 (resp. c_2) be p_1 (resp. p_2), we have $p_1 \neq p_2$. Let $X = \{1, 2, \ldots, n\} \backslash \{p_1, p_2\}$, then, by restricting all the rows of columns c_1 and c_2 in X, we get two same sub-columns. Suppose B is a matrix generated by exchanging positions of columns c_1 and c_2 in A, then B is also a share matrix in C_1. Therefore, by restricting all the rows of A and B in X, we are able to get two same sub-matrices. Namely, the

$n-2$ rows of share matrix A (the rows restricted in X) cannot uniquely determine a share matrix. Obviously, it also cannot uniquely determine a share matrix from less than $n-2$ rows. □

Example 4. In a (2,3)-VCS constructed by Droste's method, we have two basis matrices as follows:

$$M_0 = \begin{bmatrix} 010 \\ 010 \\ 010 \end{bmatrix}, M_1 = \begin{bmatrix} 100 \\ 010 \\ 001 \end{bmatrix}$$

Obviously, we can generate three (resp. six) share matrices from basis matrix M_0 (resp. M_1). When we have two rows of a share matrix, we need to uniquely determine the share matrix. By the definition of VCS, any two rows of a share matrix can reveal the secret pixel. From 1, we have the number of 1's of each column in basis matrix M_0 (resp. M_1) constructs the set $T_0 = 0, 3$ (resp. $T_1 = 1$). If the secret pixel is white (resp. black), the share matrix is constructed by permuting basis matrix M_0 (resp. M_1). Since all rows of M_0 are the same, we can uniquely determine the share matrix from its two rows if it is constructed by M_0. If the share matrix is constructed by M_1, we can also uniquely determine the share matrix from its two rows by counting the number of 1's of each column. For example, if we have two rows of a share matrix like:

$$B = \begin{bmatrix} 010 \\ 100 \end{bmatrix}$$

Because the number of 1's in each column is in the set $T_1 = 1$, we can determine the third row of the share matrix as [001]. However, if we have only one row of a share matrix, we can not determine the secret pixel, and also can not uniquely determine the share matrix. For example, if we have one row of a share matrix, like [010]. There are three share matrices, which may share the same row. These three share matrices are shown as follows.

$$B^1 = \begin{bmatrix} 010 \\ 010 \\ 010 \end{bmatrix}, B^2 = \begin{bmatrix} 010 \\ 100 \\ 001 \end{bmatrix}, B^3 = \begin{bmatrix} 010 \\ 001 \\ 100 \end{bmatrix}$$

Therefore, in a $(2, 3)$-VCS, a share matrix can be determined by any 2 rows. □

Theorem 1 presents an explicit method to uniquely determine a share matrix in C_i $(i = 0, 1)$, and in light of the above discussion, we have the following theorem:

Theorem 2. *Let $t = n - 1$, then Construction 1 generates a $(k, n-1, n)$-ESSVCS.* □

For general basis matrix visual cryptography (C_0, C_1), denote C_i^{All} as a set of all the possible columns that appear in the share matrices of C_i $(i = 0, 1)$. For any set of participants $X \subseteq P$, denote M' as a sub-matrix which is generated by restricting to the rows in X of a share matrix in C_i. *First*, we have the following lemma:

Lemma 1. *For every column c' of M', if there exists only one column $c \in C_i^{All}$ such that $c[X] = c'$, then the sub-matrix M' can uniquely determine a share matrix in C_i, where $c[X]$ is the sub-column generated by restricting to the rows in X of c.*

Proof: (Reduction to absurdity) Suppose M' cannot uniquely determine a share matrix in C_i, i.e. there exist two different share matrices, denoted by M_a and M_b, such that $M_a[X] = M_b[X] = M'$, where $M_a[X]$ is the sub-matrix generated by restricting to the rows in X of M_a. Since M_a and M_b are different share matrices, there exists at least one column that is different for M_a and M_b. Denote this column in M_a is c_a and that in M_b is c_b, i.e. $c_a \neq c_b$. Because of $M_a[X] = M_b[X]$, we have $c_a[X] = c_b[X]$, which is contradict to the assumption that there exists only one column $c \in C_i^{All}$ such that $c[X] = c'$. Hence, M' can uniquely determine a share matrix in C_i. □

According to Lemma 1, we present a general discussion for basis matrix (k, n)-VCS, denote $c_p, c_q \in C_i^{All}$ as two different columns, and denote $X_{pq}^i (\subset P)$ as the set of the participants such that for each $x \in X_{pq}^i$ satisfying $c_p[x] = c_q[x]$, where $c_p[x]$ is the x-th entry of c_p. Then we have the following theorem:

Lemma 2. *Let $t = max\{|X_{pq}^i| + 1\}$ for $p \neq q$, $1 \leq p, q \leq m$ and $i = 0, 1$, then a sub-matrix of t rows of a share matrix in C_i can uniquely determine a share matrix in C_i.*

Proof: Let c' be a column of the sub-matrix M' which is generated by restricting t rows of a share matrix in C_i $(i = 0, 1)$. Denote a set of the participants of these t rows as X, i.e. $|X| = t$, where $t = max\{|X_{pq}| + 1\}$. We prove that there only exists one column c of M such that $c[X] = c'$.

(Reduction to absurdity): Suppose there exist two columns c_a and c_b such that $c_a[X] = c_b[X] = c'$. We see that c_a and c_b have t entries with the same values, i.e. $t = |X_{ab}|$, which is impossible because $t = max\{|X_{pq}| + 1\}$ which implies $t > |X_{ab}|$.

According to Lemma 1, we have that a sub-matrix M' with t rows can uniquely determine a share matrix in C_i. □

According to Lemma 2, and let's recall that we have assumed $t \geq k$. (another reason that we assume $t \geq k$ is that, if $t < k$, then t participants cannot decide the sub-matrix of their t shares is from C_0 or C_1, and hence it may not get the ciphertext either) we hence get the following theorem immediately:

For a (k, n)-VCS, any $k - 1$ or less shares cannot get any information of the secret image. In another word, any $t(t < k)$ shares cannot decide the t-row sub-matrix is from C_0 or C_1, and hence we can not uniquely determine the share matrix. Therefore, it is reasonable to assume $t \geq k$. Further with Lemma 2, we get the following theorem:

Theorem 3. *For a basis matrix (k, n)-VCS, there exists a (k, t, n)-ESSVCS where $t = max\{k, |X_{pq}^i| + 1\}$, $p \neq q$, $1 \leq p, q \leq m$ and $i = 0, 1$.* □

According to Theorem 3, we also examined other two known constructions of (k, n)-VCS in [23, 26], and found that the two constructions both have $t = n-1$ (the same as the results in Theorem 1). Because they both take the canonical matrices as building block, where the canonical matrices mean the matrices that all the columns of a given weight occur with the same frequency. And for the canonical matrices that have a column c_i with x 1's and $n - x$ 0's where $0 < x < n$, there exists a column c_j such that only two entries are different from c_i, which implies $|X_{ij}| = n - 2$, and hence $t = n - 1$.

3.3 Bandwidth of ESSVCS scheme

We define bandwidth of the ESSVCS as the maximum amount of covert data it carries through its sub-channel. Denote columns in the basis matrix M_i as $c_1, \cdots, c_e$ and multiplicities of these columns are $a_1, \cdots, a_e$, let's recall that we have the number of share matrix in C_i being $|C_i| = \frac{(\sum_{i=1}^{e} a_i)!}{\prod_{i=1}^{e} a_i!}$ for $i \in \{0, 1\}$. To choose a share matrix in C_i, one needs at least $\log_2 |C_i|$ pseudo-random bits theoretically. By determining the share matrix which is chosen when encrypting the secret image in C_i, one can determine at most $\log_2 |C_i|$ bits information theoretically. Hence, the amount of the additional covert data that can be carried by the secret pixel i is at most $\log_2 |C_i|$ bits theoretically. We list the number of the share matrices $|C_i|$ of the VCS constructed [12] in the Table 1 and Table 2 as follows.

Actually, in practical, a pseudo-random number generator can only generate integer number of pseudo-random bits, and ciphertexts are also represented by integer number of bits. However, the values of $\log_2 |C_i|$ are rarely integers, which means that some share matrices cannot be chosen by integer number of the pseudo-random bits, and it is hard to determine all the $\log_2 |C_i|$ ciphertext

Table 1: The number of share matrices in C_0

k \ n	2	3	4	5	6	7	8	9	10
2	2	3	4	5	6	7	8	9	10
3		4!	$\frac{6!}{2!}$	$\frac{8!}{3!}$	$\frac{10!}{4!}$	$\frac{12!}{5!}$	$\frac{14!}{6!}$	$\frac{16!}{7!}$	$\frac{18!}{8!}$
4			8!	$\frac{15!}{3!2!}$	$\frac{24!}{6!3!}$	$\frac{35!}{10!4!}$	$\frac{48!}{15!5!}$	$\frac{63!}{21!6!}$	$\frac{80!}{28!7!}$
5				16!	$\frac{30!}{3!(2!)^6}$	$\frac{48!}{6!(3!)^7}$	$\frac{70!}{10!(4!)^8}$	$\frac{96!}{15!(5!)^9}$	$\frac{126!}{21!(6!)^{10}}$
6					32!	$\frac{70!}{4!(2!)^{21}3!}$	$\frac{128!}{10!(3!)^{28}6!}$	$\frac{210!}{20!(4!)^{36}10!}$	$\frac{320!}{35!(5!)^{45}15!}$
7						64!	$\frac{140!}{4!(2!)^{28}(3!)^8}$	$\frac{256!}{10!(3!)^{36}(6!)^9}$	$\frac{420!}{20!(4!)^{45}(10!)^{10}}$
8							128!	$\frac{315!}{5!(3!)^{36}(2!)^{36}4!}$	$\frac{640!}{15!(6!)^{45}(3!)^{45}10!}$
9								256!	$\frac{630!}{5!(3!)^{45}(2!)^{120}(4!)^{10}}$
10									512!

bits, hence results in wasting of the pseudo-random resources. So from the practical viewpoint, the amount of the covert data carried by the ESSVCS is impossible to reach the theoretical value.

In fact, if the secret pixels are encrypted only one at each time, in order to choose a share matrix pseudo-randomly in C_i, one needs at least $\lceil \log_2 |C_i| \rceil$ pseudo-random bits, and its length of the ciphertext can be at most $\lfloor \log_2 |C_i| \rfloor$ bits. To fully make use of the pseudo-random resources, we propose to encrypt q secret pixels at a time, i.e. the *q-pixel encryption model*. Let $q = a_0 + a_1$, where denote a_0 as the number of white pixels and a_1 as the number of black pixels, the effectiveness of using *q-pixel encryption model* rather than 1-*pixel encryption model* is as follows.

Table 2: The number of share matrices in C_1

k \ n	2	3	4	5	6	7	8	9	10
2	2!	3!	4!	5!	6!	7!	8!	9!	10!
3		4!	$\frac{6!}{2!}$	$\frac{8!}{3!}$	$\frac{10!}{4!}$	$\frac{12!}{5!}$	$\frac{14!}{6!}$	$\frac{16!}{7!}$	$\frac{18!}{8!}$
4			8!	$\frac{15!}{(2!)^5}$	$\frac{24!}{(3!)^6}$	$\frac{35!}{(4!)^7}$	$\frac{48!}{(5!)^8}$	$\frac{63!}{(6!)^9}$	$\frac{80!}{(7!)^{10}}$
5				16!	$\frac{30!}{3!(2!)^6}$	$\frac{48!}{6!(3!)^7}$	$\frac{70!}{10!(4!)^8}$	$\frac{96!}{15!(5!)^9}$	$\frac{126!}{21!(6!)^{10}}$
6					32!	$\frac{70!}{(3!)^7(2!)^7}$	$\frac{128!}{(6!)^8(3!)^8}$	$\frac{210!}{(10!)^9(4!)^9}$	$\frac{320!}{(15!)^{10}(5!)^{10}}$
7						64!	$\frac{140!}{4!(2!)^{28}(3!)^8}$	$\frac{256!}{10!(3!)^{36}(6!)^9}$	$\frac{420!}{20!(4!)^{45}(10!)^{10}}$
8							128!	$\frac{315!}{(4!)^9(2!)^{84}(3!)^9}$	$\frac{640!}{(10!)^{10}(3!)^{120}(6!)^{10}}$
9								256!	$\frac{630!}{5!(3!)^{45}(2!)^{120}(4!)^{10}}$
10									512!

First: the number of pseudo-random bits required to choose the share matrices when the *q-pixel encryption model* is $\lceil a_0 \log_2 |C_0| + a_1 \log_2 |C_1| \rceil$, and it satisfies:

$$\lceil a_0 \log_2 |C_0| + a_1 \log_2 |C_1| \rceil \leq a_0 \lceil \log_2 |C_0| \rceil + a_1 \lceil \log_2 |C_1| \rceil \tag{5}$$

which implies less pseudo-random bits are required by using the *q-pixel encryption model* than the 1-*pixel encryption model*.

Second: the number of pseudo-random bits determined by the share matrices when encrypting q secret pixels at each time is $\lfloor a_0 \log_2 |C_0| + a_1 \log_2 |C_1| \rfloor$, and it satisfies:

$$\lfloor a_0 \log_2 |C_0| + a_1 \log_2 |C_1| \rfloor \geq a_0 \lfloor \log_2 |C_0| \rfloor + a_1 \lfloor \log_2 |C_1| \rfloor \tag{6}$$

which implies more pseudo-random bits can be determined by using the *q-pixel encryption model* than the 1-*pixel encryption model*.

A problem for the *q-pixel encryption model* is that, when encrypting more secret pixels at a time, the encryption scheme becomes more complex. So there exists a trade-off for the value of q.

To make things clear, we present the following example for a $(2, 2, 3)$-ESSVCS:

Example 5. For the sets

$$C_0 = \left\{ \begin{bmatrix} 100 \\ 100 \\ 100 \end{bmatrix}, \begin{bmatrix} 010 \\ 010 \\ 010 \end{bmatrix}, \begin{bmatrix} 001 \\ 001 \\ 001 \end{bmatrix} \right\} \tag{7}$$

$$C_1 = \left\{ \begin{bmatrix} 100 \\ 010 \\ 001 \end{bmatrix}, \begin{bmatrix} 100 \\ 001 \\ 010 \end{bmatrix}, \begin{bmatrix} 010 \\ 100 \\ 001 \end{bmatrix}, \begin{bmatrix} 010 \\ 001 \\ 100 \end{bmatrix}, \begin{bmatrix} 001 \\ 100 \\ 010 \end{bmatrix}, \begin{bmatrix} 001 \\ 010 \\ 100 \end{bmatrix} \right\} \tag{8}$$

We have that, from theoretic point of view, the amount of information bits that can be carried by a white secret pixel is $\log_2 |C_0| = \log_2 3$ and by a black secret pixel is $\log_2 |C_1| = \log_2 6$. And for 10 secret pixels with 5 white secret pixels and 5 black secret pixels the value will be $5 \log_2 3 + 5 \log_2 6 \approx 20.85$.

However, in practical, the 10-*pixel encryption model*, where take $a_0 = 5$ and $a_1 = 5$ as example, we have the amount of information that can be carried is $\lfloor \log_2 3^5 + \log_2 6^5 \rfloor = 20$, which is more than 1-*pixel encryption model*, where the corresponding value is $5\lfloor \log_2 3 \rfloor + 5\lfloor \log_2 6 \rfloor = 15$. □

At this point, we can calculate the bandwidth of the ESSVCS as follows:

Theorem 4. *For a secret image S_I which consists of n_w white pixels and n_b black pixels, the bandwidth W of the ESSVCS is $W = \lfloor n_w \log_2 |C_0| + n_b \log_2 |C_1| \rfloor$, and it is achieved when using the* q_a-pixel encryption model *where $q_a = n_w + n_b$.*

Proof: For the *q_a-pixel encryption model* where $q_a = n_w + n_b$, which implies encrypt all the secret pixels in the secret image at each time. And it is clear that the amount of covert data carried by such ESSVCS is $W = \lfloor n_w \log_2 |C_0| + n_b \log_2 |C_1| \rfloor$. We only need to prove that W reaches its maximum when using the *q_a-pixel encryption model*, i.e. if one divides all the pixels in the secret image into several parts, and encrypts these parts respectively, the amount of covert data carried is less than the *q_a-pixel encryption model*.

Without loss of generality, let $q_a = q_1 + q_2$ (i.e. divide into two parts) and suppose encryption of the secret image S_I is realized by using *q_1-pixel encryption model* and *q_2-pixel encryption model*, and let $q_1 = a_0 + a_1$, $q_2 = b_0 + b_1$, where a_0, b_0 are the number of white pixels and a_1, b_1 are the number of black pixels. We have that the total number of pseudo-random bits can be determined is $\lfloor a_0 \log_2 |C_0| + a_1 \log_2 |C_1| \rfloor + \lfloor b_0 \log_2 |C_0| + b_1 \log_2 |C_1| \rfloor$, which is not greater than $\lfloor (a_0 + b_0) \log_2 |C_0| + (a_1 + b_1) \log_2 |C_1| \rfloor = \lfloor n_w \log_2 |C_0| + n_b \log_2 |C_1| \rfloor$. Hence, the theorem is true. □

3.4 On decoding the ciphertext

For ESSVCS, in order to encrypt the secret pixels and decode the ciphertext, one needs to set a bijection between the set of pseudo-random numbers (ciphertext) and the set of share matrices. A simple way to realize that is to generate a table which contains all the share matrices and their corresponding random numbers. When the dealer generates the shares, (s)he needs to generate a pseudo-random number and find the corresponding share matrix by table-lookup, then (s)he can

encrypt the shares by using the share matrix. When decoding the ciphertext, the participants get the share matrices according to the Theorem 1, and find the corresponding numbers by table-lookup, hence, they get the ciphertext. Disadvantage of this decoding method is that, the table requires us store all the share matrices in sets C_0 and C_1, and hence it has large memory requirements. In this subsection, we propose a decoding method which is more efficient than the above mentioned method.

The proposed decoding method contains two subroutines: the first is $MTN(S)$, which takes a share matrix in C_i $(i = 0, 1)$ as its input and generates a number between 1 and $m!$, the second is $NTM(N)$, which takes a number between 1 and $m!$ as its input and generates a share matrix S. The subroutines $MTN(S)$ and $NTM(N)$ form a bijection between the set of the share matrices and the set of numbers between 1 and $m!$.

By using $MTN(S)$ and $NTM(N)$, when the dealer encrypts a secret pixel p, (s)he first generates a pseudo-random number between 1 and $m!$, and then consults the subroutine $NTM(N)$ to generate a share matrix in C_i $(i = 0, 1)$, and encrypts the secret pixel p by using the share matrix. When the participants decode the ciphertext, they first generate the share matrix according to Theorem 1, and consult the subroutine $MTN(S)$ to get the ciphertext.

Denote the columns of the basis matrix as $c_1, \cdots, c_m$, first we take the case that $c_1, \cdots, c_m$ are pairwise different into consideration. In this part, we treat a matrix as a set of columns. The subroutine $MTN(S)$ which outputs a number between 1 and $m!$ given a share matrix S as its input is:

Subroutine: MTN(S)

For $i = 1$ to $m - 1$
  Find c_i in S, assume that c_i is the J_i-th column of S
  Delete c_i from S
Output $N = 1 + \sum_{i=1}^{m-1} ((m - i)!)(J_i - 1)$

The subroutine $NTM(N)$ which outputs a share matrix S given a number between 1 and $m!$ as its input is:

Subroutine: NTM(N)

Initial S as an empty matrix
$N_0 \leftarrow N - 1$
For $i = 1$ to $m - 1$
  $J_i \leftarrow \lfloor \frac{N_{i-1}}{(m-i)!} \rfloor + 1$
  $N_i \leftarrow N_{i-1} - (J_i - 1)((m - i)!)$
Insert c_m to S as its 1-st column
For $i = m - 1$ to 1
  Insert column c_i into S as its J_i-th column
Output S

According to the subroutines $MTN(S)$ and $NTM(N)$ above, we have the following theorem:

Theorem 5. *The subroutines $MTN(S)$ and $NTM(N)$ form a bijection between the set of share matrices in C_i $(i = 0, 1)$ and the set of numbers between 1 and $m!$.*

Proof: Because in subroutines $MTN(S)$ and $NTM(N)$, we represent the share matrices by the positions of its columns $(J_1, J_2, \cdots, J_{m-1})$ where $1 \leq J_i \leq m+1-i$ for $i = 1, 2, \cdots, m-1$, we only need to prove that $MTN(S)$ and $NTM(N)$ form a bijection between the sets $X = \{(J_1, J_2, \cdots, J_{m-1}) | 1 \leq J_i \leq m + 1 - i \ for \ i = 1, 2, \cdots, m - 1\}$ and $Y = \{1, 2, \cdots, m!\}$. Denote $f : X \to Y$ as a map from X to Y, we prove that f is a bijection.

First, given a number in Y, according to $NTM(N)$, there exists a $(J_1, J_2 \cdots, J_{m-1})$, hence f is a surjection.

Second, for any two different elements in X, $J = (J_1, J_2, \cdots, J_{m-1})$ and $J' = (J'_1, J'_2, \cdots, J'_{m-1})$ such that $J \neq J'$, we prove that their corresponding numbers $f(J)$ and $f(J')$ are different.

According to $MTN(S)$, we have $f(J) = 1 + \sum_{i=1}^{m-1}((m-i)!)(J_i - 1)$ and $f(J') = 1 + \sum_{i=1}^{m-1}((m-i)!)(J'_i - 1)$. Denote i^* as the smallest number that $J_{i^*} \neq J'_{i^*}$, without loss of generality, we suppose $J_{i^*} > J'_{i^*}$, i.e. $J_{i^*} - J'_{i^*} \geq 1$. Thus, we have:

$$\begin{aligned} f(J) - f(J') =& \sum_{i=1}^{m-1}((m-i)!)(J_i - J'_i) \\ =&(m-i^*)!(J_{i^*} - J'_{i^*}) + \sum_{i=i^*+1}^{m-1}((m-i)!)(J_i - J'_i) \\ \geq& (m-i^*)! + \sum_{i=i^*+1}^{m-1}((m-i)!)(J_i - J'_i) \end{aligned}$$

Because $1 \leq J_i, J'_i \leq m+1-i$, we have $-(m-i) \leq J_i - J'_i \leq m-i$, hence

$$\begin{aligned} f(J) - f(J') \geq& (m-i^*)! - \sum_{i=i^*+1}^{m-1}((m-i)!)(m-i) \\ =&(m-i^*)! - ((m-i^*)! - 1) \\ =&1 \end{aligned}$$

Therefore, $f(J) - f(J') \neq 0$, we have f is an injection. Hence, f is a bijection and the theorem follows. □

Example 6. For a (2,3)-VCS, the basis matrix M_1 has three different columns. M_1 and its three columns c_1, c_2, c_3 are shown as follows:

$$M_1 = \begin{bmatrix} 100 \\ 010 \\ 001 \end{bmatrix}, c_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, c_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, c_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

By subroutines $MTN(S)$ and $NTM(N)$, we can construct a bijection between the set of share matrices generated by M_1 and the set of numbers between 1 and 3!. The detailed bijection can be shown as follows.

$$No.1: \begin{bmatrix} 100 \\ 010 \\ 001 \end{bmatrix}, No.2: \begin{bmatrix} 100 \\ 001 \\ 010 \end{bmatrix}, No.3: \begin{bmatrix} 010 \\ 100 \\ 001 \end{bmatrix}$$

$$No.4: \begin{bmatrix} 001 \\ 100 \\ 010 \end{bmatrix}, No.5: \begin{bmatrix} 010 \\ 001 \\ 100 \end{bmatrix}, No.6: \begin{bmatrix} 001 \\ 010 \\ 100 \end{bmatrix}$$

□

For the case that there are identical columns in the basis matrix, which means that there are identical share matrices in the $m!$ permutations of the basis matrix. Suppose there are e different columns in the basis matrix, and the multiplicities of these columns are $a_1, a_2, \cdots, a_e$. Denote N_d as the number of the different share matrices in C_i, then we have $N_d = \frac{(\sum_{i=1}^{e} a_i)!}{\prod_{i=1}^{e} a_i!}$, for $i \in \{0, 1\}$. Each share matrix appears $\frac{m!}{N_d}$ times in the $m!$ permutations.

Furthermore, according to the subroutine $MTN(S)$, each permutation corresponds to a number between 1 and $m!$, we can divide these $m!$ numbers into N_d groups, where each group contains $\frac{m!}{N_d}$ numbers, and the numbers in one group correspond to an identical share matrix. We hence can form an array of length N_d by choosing the smallest number of each group. Denote this array as A, and denote $S_1^i, S_2^i \cdots, S_{N_d}^i$ as all the different share matrices in the set C_i, the following subroutine generates A:

Subroutine: MC

Initial an empty array A
For $j = 1$ to N_d
  For $q = 1$ to m
    Find the first c_q in S_j^i from left to right, assume that c_q is the J_q-th column of S_j^i
    Delete c_q from S_j^i
  $A[j] \leftarrow 1 + \sum_{q=1}^{m-1}((m-q)!)(J_q - 1)$

To differentiate the two cases whether there exist and do not exist identical columns, we denote MTN-d(S) and NTM-d(N) as the corresponding subroutines for the case that there exist identical columns:

Subroutine: MTN-d(S)

$A \leftarrow MC$
For $q = 1$ to m
  Find the first c_q in S_j^i from left to the right, assume that c_q is the J_q-th column of S_j^i
  Delete c_q from S_j^i
$N' \leftarrow 1 + \sum_{q=1}^{m-1}((m-q)!)(J_q - 1)$
For $r = 1$ to N_d
  if $A[r] = N'$
    Output r

Subroutine: NTM-d(N)

$A \leftarrow MC$
$N' \leftarrow A[N]$
$S \leftarrow NTM(N')$
Output S

According to the Theorem 5, we have that, each group only has one smallest number. Hence the array A is a bijection from the set $\{1, 2, \cdots, N_d\}$ and the set of the smallest numbers in each group. Furthermore, because each group corresponds to a different share matrix we have that the MTN-d(S) and NTM-d(N) form a bijection between the set $\{1, 2, \cdots, N_d\}$ and the set of share matrices $\{S_1^i, S_2^i \cdots, S_{N_d}^i\}$. We summarize this result as the following theorem:

Theorem 6. *The subroutines MTN-$d(S)$ and NTM-$d(N)$ form a bijection between the set of share matrices in C_i ($i = 0, 1$) and the set of numbers between 1 and N_d.* □

Example 7. For a (2,3)-VCS, there are identical columns in basis matrix M_0. M_0 and its two different columns c_1 and c_2 with multiplicities 1 and 2 are shown as follows.

$$M_1 = \begin{bmatrix} 100 \\ 100 \\ 100 \end{bmatrix}, c_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, c_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

By subroutines $MTN - d(S)$ and $NTM - d(N)$, we can construct a bijection between the set of share matrices generated by M_0 and the set of numbers between 1 and 3. The detailed bijection can be shown as follows.

$$No.1 : \begin{bmatrix} 100 \\ 100 \\ 100 \end{bmatrix}, \; No.2 : \begin{bmatrix} 100 \\ 010 \\ 001 \end{bmatrix}$$

□

The above subroutines are more efficient than the simple table-lookup method. Particularly, for the case that the columns $c_1, c_2, \cdots, c_m$ are pairwise different, the subroutines $MTN(S)$ and $NTM(N)$ are efficient, because they only need fixed memory requirements. For the case that there are identical columns in $c_1, c_2, \cdots, c_m$, the memory requirement of the subroutines $MTN - d(S)$ and $NTM - d(N)$ relates to the value of m. Because they only need to store the indexes of the share matrices $A[1], A[2], \cdots, A[N_d]$, they are more efficient than the simple table-lookup method. Furthermore, the table (the array A in Subroutine MC) can be previously generated and reusable.

4 Comparisons of ESSVCS and TiOISSS

From the viewpoint of carrying amount of the secret, both the ESSVCS and the TiOISSS are computer aided and carry two types of secrets, one is a secret image that can be revealed by stacking the shares, and the other is covert data which is revealed by computation. The three TiOISSS schemes [18–20] can be also treated as (k, k, n)-TiOISSS, which means a vague secret image is revealed by stacking any k out of n shares, and further a much finer gray-scale secret image (i.e. the covert data) is revealed by computation with these k shares.

Taking the information carrying capability into consideration, we compare the amount of covert data carried by the ESSVCS and three TiOISSSs [18–20].

First, the covert data carried by the ESSVCS is greater than that in Lin et al.'s TiOISSS [18]. Bandwidth of the proposed ESSVCS has been discussed in Theorem 4, and it can be evaluated from Table 1 and Table 2. Lin et al.'s TiOISSS [18] groups the share matrices into different types to carry covert data according to the first row. Let m be pixel expansion of the basis matrix, where each row contains b '1' and w '0' and m=b+w. There are $\binom{m}{w}$ different types of share matrix, and each secret pixel in VCS carries $\log_2 \binom{m}{w}$ bits. We list the number of share matrices generated by Droste [12] with different types in Table 3.

Note that in order to satisfy the security, we can only choose the type of one row and the remaining $(n-1)$ rows are then determined according to the type of share matrix. Therefore, only $1/n$ part of each share can be used to carry $\log_2 \binom{m}{w}$ bits. Since the covert data of each share is taken from the shadow image generated by polynomial-based secret sharing scheme, the total secret information carried by VCS is $k \cdot |S_I| \cdot \log_2 \binom{m}{w} / n$ bits, where S_I is the binary secret image of VCS.

Table 3: The number of share matrices with different types in Lin et al.'s TiOISSS

k \ n	2	3	4	5	6	7	8	9	10
2	2	3	4	5	6	7	8	9	10
3		$\frac{4!}{2!2!}$	$\frac{6!}{3!3!}$	$\frac{8!}{4!4!}$	$\frac{10!}{5!5!}$	$\frac{12!}{6!6!}$	$\frac{14!}{7!7!}$	$\frac{16!}{8!8!}$	$\frac{18!}{9!9!}$
4			$\frac{8!}{4!4!}$	$\frac{15!}{9!6!}$	$\frac{24!}{16!8!}$	$\frac{35!}{25!10!}$	$\frac{48!}{36!12!}$	$\frac{63!}{49!14!}$	$\frac{80!}{64!16!}$
5				$\frac{16!}{8!8!}$	$\frac{30!}{15!15!}$	$\frac{48!}{24!24!}$	$\frac{70!}{35!35!}$	$\frac{96!}{48!48!}$	$\frac{126!}{63!63!}$
6					$\frac{32!}{16!16!}$	$\frac{70!}{40!30!}$	$\frac{128!}{80!48!}$	$\frac{210!}{140!70!}$	$\frac{320!}{224!96!}$
7						$\frac{64!}{32!32!}$	$\frac{140!}{70!70!}$	$\frac{256!}{128!128!}$	$\frac{420!}{210!210!}$
8							$\frac{128!}{64!64!}$	$\frac{315!}{175!140!}$	$\frac{640!}{384!256!}$
9								$\frac{256!}{128!128!}$	$\frac{630!}{315!315!}$
10									$\frac{512!}{256!256!}$

Second, there is no fixed relationship between the amount of covert data carried by the ESSVCS and that of Yang et al.'s TiOISSS [19]. The TiOISSS [19] replaces the black pixels in the shares with gray pixels generated by polynomial-based secret sharing scheme. Therefore, each row of share matrix carries 8 bits, and the total amount of covert data are $8000 \cdot |S_I|$ bits, where b is the number of '1' in each row of share matrix. In most cases, especially when n is a small number, the covert data carried by Yang et al.'s TiOISSS [19] is more than that in the ESSVCS.

However, in some cases, the ESSVCS can carry more data. For example, for a (2,10)-VCS constructed by Droste [12], we have $\log_2 |C_0| = 3.32$, $\log_2 |C_1| = 21.79$, and $b = 1, k = 2$. In the ESSVCS, each white secret pixel carries 3.32 bits and each black secret pixel carries 21.79 bits, while in Yang et al.'s TiOISSS [19], sharing one secret pixel carries 16 bits. With proper proportion of the numbers of white secret pixels to black secret pixels, the ESSVCS can carry more covert data than Yang et al.'s TiOISSS [19].

Li et al.'s TiOISSS [20] improved Yang et al.'s scheme [19] by restricting the gray values of share pixels. The visual quality of the revealed image can be improved by increasing the share size. Hence each share carries less information than that in Yang et al.'s scheme [19].

From viewpoint of visual quality, both ESSVCS and TiOISSS can visually recover the secret image by stacking shares. The ESSVCS and Lin et al.'s TiOISSS [18] used traditional VCS as the building block, hence the recovered secret image is as same as that of the traditional VCS. In Yang et al.'s TiOISSS [19] and Li et al.'s TiOISSS [20], the black pixels of shares are replaced by gray pixels, and the contrast of VCS is diminished. Therefore, visual quality of the recovered secret image by stacking shares is deteriorated in Yang et al.'s TiOISSS [19] and Li et al.'s TiOISSS [20], which is a disadvantage of their scheme. However, in Li et al.'s TiOISSS [20], the visual quality can be improved with the cost of larger share size.

Besides, another disadvantage of Yang et al.'s TiOISSS scheme [19] and Li et al.'s TiOISSS [20] is that, to reconstruct the covert data the participants have to obtain the greyness of each sub-pixel precisely, which is impractical if the shares are printed on transparencies. Occasional scrub may change the greyness of sub-pixels in the transparencies, which will be impossible to reconstruct the covert data.

Both the ESSVCS and Lin et al.'s TiOISSS [18] carry covert data by choosing different share matrices, hence there is a bijection between the set of pseudo-random numbers (ciphertext) and the set of share matrices. In Lin et al.'s TiOISSS [18], it employs a lookup table to map the different types of share matrices to the set of pseudo-random numbers. The disadvantage of their scheme is that, the table needs to store all the types of share matrices, which has large memory requirements. However, in the ESSVCS, an efficient algorithm is introduced to make the mapping more convenient.

5 Conclusions

In this paper, we proposed a construction of the (k, t, n)-ESSVCS scheme, which can carry additional covert data compared to the traditional (k, n)-VCS scheme by treating the pseudo-random inputs as a sub-channel. We analyzed some issues related to ESSVCS scheme such as the pseudo-randomness that the input of VCS requires, sufficient conditions to uniquely determine a share in the set C_i $(i = 0, 1)$, and bandwidth of the proposed ESSVCS scheme. We also presented an efficient algorithm to decode ESSVCS secret. At last, comparisons of some relevant VCS schemes are given such as the TiOISSS scheme [18–20].

The proposed (k, t, n)-ESSVCS scheme is especially useful for the case $(n-1, n-1, n)$-ESSVCS and the case (n, n, n)-ESSVCS, because in these cases, the qualified participants could get secret and covert data simultaneously. The constructions of $(k, n-1, n)$-ESSVCS and (k, n, n)-ESSVCS can be easily implemented by the proposed scheme. For general value of $k < t < n-1$, we left it as an open problem for future study.

Our (k, t, n)-ESSVCS scheme is not so perfect, there are more space to improve our work in future. We will provide a more suitable solution for concanating the keys to each share.

It is well known that the shares of VC are random. The content properties of random shares have not been fully explored, this could reveal wasteful use of space and can easily lead an adversary to suspect the presence of a hidden message. We will study the space complexity of a VC share in future. We plan on using cipher text to generate the master key and presenting the resulting encrypted information in black/white blocks. The content-based VC shares will fully use the available key space.

Acknowledgements

The paper is submitted to the Springer Transactions on Data Hiding and Multimedia Security (DHMS) on 28 Feb 2013, partial work has been published in the proceedings of International Workshop on Digital-forensics and Watermarking 2012 (IWDW2012).

This work was supported by NSFC grant (No. 60903210), the "Strategic Priority Research Program" of the Chinese Academy of Sciences (No. XDA06010701) and the Cryptography Research Project (No. Y3Z001B102). Thanks for the anonymous reviewers' invaluable constructive comments and suggestions.

Bibliography

[1] M. Naor and A. Shamir. Visual cryptography. In *EUROCRYPT '94, Springer-Verlag Berlin*, volume LNCS 950, pages 1–12, 1995.

[2] B. Surekha, G. Swamy, and K.S. Rao. A multiple watermarking technique for images based on visual cryptography. In *Computer Applications*, volume 1, pages 77–81, 2010.

[3] T. Monoth and B. Anto P. Tamperproof transmission of fingerprints using visual cryptography schemes. In *Procedia Computer Science*, volume 2, pages 143–148, 2010.

[4] J. Weir and W. Yan. Resolution variant visual cryptography for street view of google maps. In *Proceedings of the ISCAS*, pages 1695–1698, 2010.

[5] C.N. Yang, T.S. Chen, and M.H. Ching. Embed additional private information into two-dimensional bar codes by the visual secret sharing scheme. In *Integrated Computer-Aided Engineering*, volume 13, Number 2, pages 189–199, 2006.

[6] J. Weir and W. Yan. A comprehensive study of visual cryptography. In *Springer Transactions on Data Hiding and Multimedia Security*, volume 6010, pages 70–105, 2010.

[7] S. Cimato and C.N. Yang. *Visual cryptography and secret image sharing.* CRC Press, Taylor & Francis, 2011.

[8] J. P. Weir and WeiQi Yan. *Visual Cryptography and Its Applications.* Ventus Publishing Aps, 2012.

[9] F. Liu, C.K. Wu, and X.J. Lin. The alignment problem of visual cryptography schemes. In *Designs, Codes and Cryptography*, volume 50, pages 215–227, 2009.

[10] W.Q. Yan, D. Jin, and M.S. Kankanhalli. Visual cryptography for print and scan applications. In *Proceedings of the 2004 International Symposium on Circuits and Systems*, volume 5, pages 572–575, 2004.

[11] M. Iwamoto and H. Yamamoto. A construction method of visual secret sharing schemes for plural secret images. In *IEICE Transactions on Fundamentals*, volume E86-A.NO.10, pages 2577–2588, 2003.

[12] S. Droste. New results on visual cryptography. In *CRYPTO '96, Springer-Verlag Berlin LNCS*, volume 1109, pages 401–415, 1996.

[13] M. Iwamoto, W. Lei, K. Yoneyama, N. Kunihiro, and K. Ohta. Visual secret sharing schemes for multiple secret images allowing the rotation of shares. In *IEICE Transactions on Fundamentals*, volume E89-A. NO.5, pages 1382–1395, 2006.

[14] C.N. Yang and C.S. Laih. New colored visual secret sharing schemes. In *Designs, Codes and Cryptography*, volume 20, pages 325–335, 2000.

[15] D. Jin, W.Q. Yan, and M.S. Kankanhalli. Progressive color visual cryptography. In *Journal of Electronic Imaging*, volume 14, Issue 3, page 033019, 2005.

[16] S.J. Shyu. Efficient visual secret sharing scheme for color images. In *Pattern Recognition*, volume 39, pages 866–880, 2006.

[17] A. Shamir. How to share a secret. In *Communications of the ACM*, volume 22 (11), pages 612–613, 1979.

[18] S.J. Lin and J.C. Lin. VCPSS a two in one two decoding options image sharing method combining visual cryptography (VC) and polynomial style sharing PSS approaches. In *Pattern Recognition*, volume 40, pages 3652–3666, 2007.

[19] C.N. Yang and C.B. Ciou. Image secret sharing method with two-decoding-options: Lossless recovery and previewing capability. In *Image and Vision Computing*, volume 28, pages 1600–1610, 2010.

[20] P. Li, P.J. Ma, X.H. Su, and C.N. Yang. Improvements of a two-in-one image secret sharing scheme based on gray mixing model. In *Journal of Visual Communication and Image Representation*, volume 23, Issue 3, pages 441–453, 2012.

[21] W.P. Fang and J.C. Lin. Visual cryptography with extra ability of hiding confidential data. In *Journal of Electronic Imaging*, volume 15(2), page 023020, 2006.

[22] C. Blundo, A. De Santis, and D.R. Stinson. On the contrast in visual cryptography schemes. In *Journal of Cryptology*, volume 12(4), pages 261–289, 1999.

[23] H. Koga. A general formula of the (t,n)-threshold visual secret sharing scheme. In *ASIACRYPT '2002, Springer-Verlag LNCS*, volume 2501, pages 328–345, 2002.

[24] W.F. Ehrsam, C.H.W. Meyer, J.L. Smith, and W.L. Tuchman. *Message verification and transmission error detection by block chaining.* 1976.

[25] J. Soto and L. Bassham. Randomness testing of the advanced encryption standard finalist candidates. In *Proceedings AES3, New York, http://csrc.nist.gov/publications/nistir/ir6483.pdf*, 2001.

[26] C. Blundo, A. De Bonis, and A. De Santis. Improved schemes for visual cryptography. In *Designs, Codes and Cryptography*, volume 24, pages 255–278, 2001.